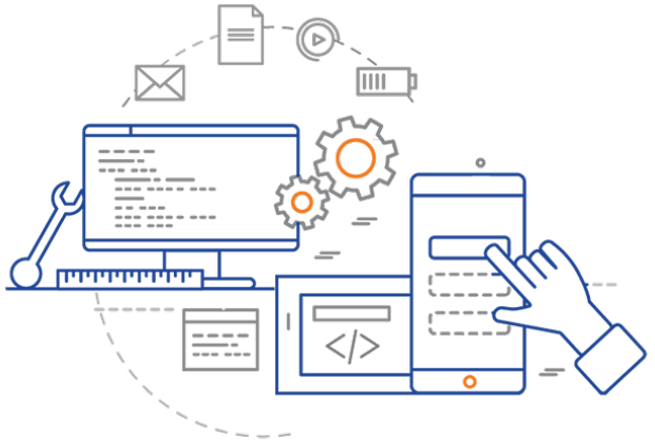


CS 398: Application Development

Pair Programming

Motivation; Styles of pairing; Setup

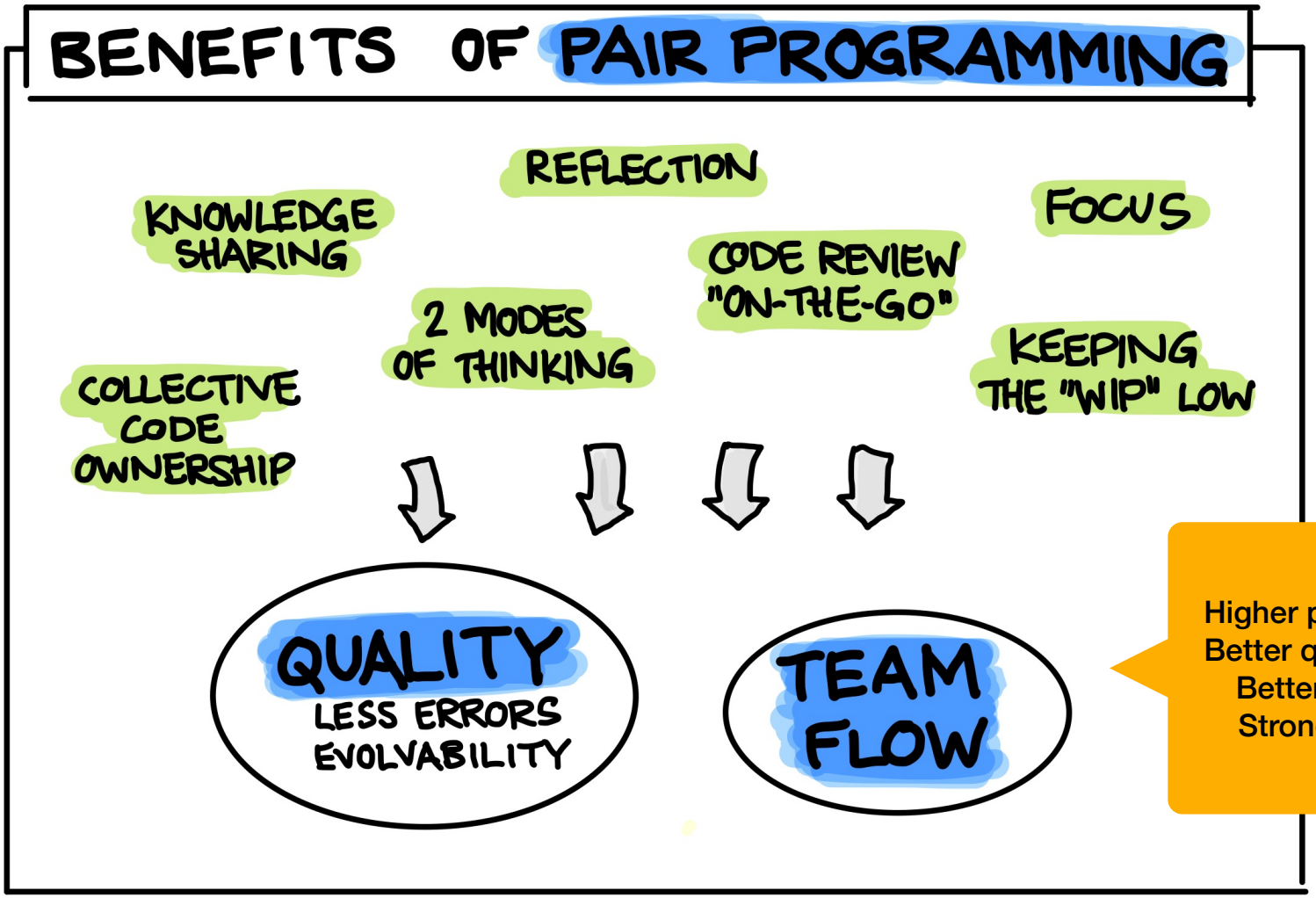
What is pair programming?

Pair programming means that two people work together to design and implement code.

One machine, two team members.

- This is a very collaborative way of working that involves a lot of communication and collaboration between them.
- While a pair of developers work on a task together, they do not only write code, they also plan and discuss their work. They clarify ideas on the way, discuss approaches and come to better solutions.

This was an *Extreme Programming* practice that has gone mainstream.



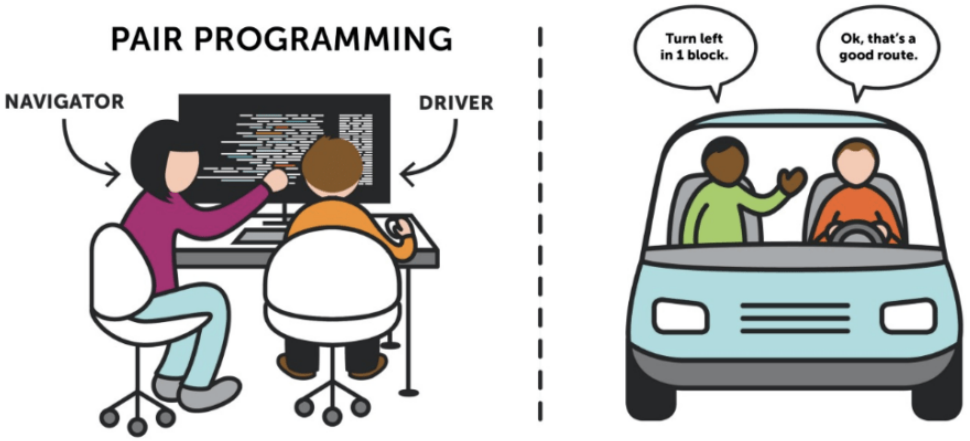
https://martinfowler.com/articles/on-pair-programming.html#HowToPair

Style: Driver and Navigator

This is the classic form of pair programming.

The **Driver** is the person at the wheel, i.e. the keyboard. She is focussed on completing the tiny goal at hand, ignoring larger issues. A driver should always talk through what she is doing while doing it.

The **Navigator** is in the observer position. She reviews the code on-the-go, gives directions and shares thoughts. The navigator also has an eye on the larger issues, bugs, and makes notes of potential next steps or obstacles.



A team can accomplish more, easier, than people working alone.

How do you work together?

A common flow goes like this:

- Start with a reasonably well-defined task. Pick a requirement or user story for example.
- Write down your goals and keep track of them (paper, whiteboard, sticky notes).
- Agree on one small goal at a time. This can be defined by a unit test, or by a commit message, or some goal that you've written down.
- Discuss the overall design as a team. The driver then implements it while the navigator watches.

As the **navigator**, you are in charge of medium-term thinking and planning ahead.

- Provide feedback on the immediate goal. Anything else should be written down for afterwards.

As the **driver**, focus on the immediate goal.

- You are coding. Explain what you are doing to the navigator. Ask for feedback if you need it.

Other Styles: Ping-Pong Coding

This technique is ideal when you have a clearly defined task that can be implemented in a test-driven way.

- “Ping”: Developer A writes a failing test
- “Pong”: Developer B writes the implementation to make it pass.
- Developer B then starts the next “Ping”, i.e. the next failing test.
- Each “Pong” can also be followed by refactoring the code together, before you move on to the next failing test.

We’ll talk more about TDD in the Unit Testing lecture.

Other Styles: Strong-Style Pairing

In this style, the navigator is much more experienced (with the language, the tool, the codebase). Our goal isn't so much collaboration as it is knowledge-transfer between the team members.

- **Navigator**: the more experienced person, who guides the driver, and explicitly tells them what to do (i.e. what to code, what changes to make).

- **Driver**: the less experienced person, who writes the code that the navigator tells them to write. They take explicit direction from the navigator.

The driver should trust the navigator and be "comfortable with incomplete understanding". Challenges to the solution should be discussed after the implementation session.

You may encounter this style in the workplace, but it's unlikely that we will use it here.

Switching Roles

Switching roles while pairing is essential to the process. Each one of you should take turns driving and navigating.

- Plan to switch at natural break points (maybe after a change is completed and tested).
- Ask first. If you feel a need to switch your role, then ask and explain why. Remember, collaboration.
    - Do not grab the keyboard. Ask first.
    - Do not push the keyboard at the navigator. Ask first.

Setup

Physical Setup

Ideally, you would work together in the same space.

- Make sure both of you have enough space, and that both of you can sit facing the computer.
- Agree on the computer setup (key bindings, IDE etc). Check if your partner has any particular needs (e.g. larger font size, higher contrast, …)

Remote Setup

There are also development tools that are designed to specifically support remote sharing:

- JetBrains Code With Me: https://www.jetbrains.com/code-with-me/
- Visual Studio Live Share: https://visualstudio.microsoft.com/services/live-share/
- Codeshare: https://codeshare.io